1. **Based on what we know about linked lists, stacks, and queues, design a queue data structure:**
2. **What functions are we likely to need for a queue to function like the one discussed in class?**

**Answer:** Following are the functions which are required for queue.

* Enqueue
  + It will allow to add item in the queue.
  + It will be FIFO structure which means First In First Out.
  + It has Front and Rear for Items
* Dequeue
  + It will allow to remove item(s) in the queue
  + It will be FIFO structure which means First In First Out.
* Is\_Full
* Is\_Empty

1. **What values will we need to know about the structure for our queue to function properly?**

**Answer:** We need to know the first and last element of the queue for the queue to function properly.

1. **Based on what we know about linked lists, design a list data structure that allows us to add (insert) or remove (delete) values at a given location in the list (instead of the top of a stack or the front or back of a queue):**
2. **What functions are we likely to need for a list to function like this?**

**Answer:** Following are the functions which will be required for linked list to function properly.

* InsertAfter(val x, val key) ; Here, val is a template datatype
* InsertBefore(val x, val key)

(In these functions, ‘key’ will be inserted to the linked list before or after the node which contains ‘x’ as its data.)

1. **What values will we need to know about the structure for our list to function properly?**

**Answer:** Following are the values which we need to know for our list to function properly.

* + Head of linked list(start)
  + Tail of linked list(end)
  + Next pointer for each node